Purdue University
Purdue e-Pubs

Department of Computer Science Technical Reports
Department of Computer Science

1986

Automatic Rational Parameterization of Curves and Surfaces I: Conics and Conicoids

Shreeram S. Abhyankar
Purdue University, ram@cs.purdue.edu

Chanderjit Bajaj

Report Number:
86-583

AUTOMATIC RATIONAL PARAMETERIZATION OF CURVES AND SURFACES I: CONICS AND CONICOIDS

Shreeram S. Abhyankar
Chanderjit Bajaj

CSD-TR-583
March 1986

Automatic Rational Parameterization of Curves and Surfaces I: Conics and Conicoids

Shreeram S. Abhyankar[†] and *Chanderjit Bajaj*

Department of Computer Science,
Purdue University,
West Lafayette, IN 47907.

Abstract

We describe algorithms to obtain rational parametric equations, (a polynomial equation divided by another), for degree two curves (conics) and degree two surfaces (conicoids), given the implicit equations. We further consider the rational parameterizations over the fields of rationals, reals and complex numbers. In doing so, solutions are given to important subproblems of finding rational, real or complex points on the given conic curve or conicoid surface. Polynomial parameterizations are obtained whenever they exist for the conics and conicoids. These algorithms have been implemented on a VAX-780 using VAXIMA.

† Department of Mathematics, Purdue University, West Lafayette, IN

1. Introduction

General curves and surfaces can be represented by parametric or implicit equations. For various reasons related to efficient computability, (greater freedom in controlling shape and direct ease in performing transformations), the dominant means of representing curves and surfaces in geometric modeling is the parametric equation [4]. A general (degree two) conic implicit equation is given by $I(x,y) = ax^2 + by^2 + cxy + dx + ey + f = 0$, and rational parametric equations given by $x = p(t)/q(t)$ and $y = k(t)/l(t)$, where p, q, k and l are univariate polynomials. Further a general (degree two) conicoid implicit equation is given by $I(x,y,z) = ax^2 + by^2 + cz^2 + dxy + exz + fyz + gx + hy + iz + j = 0$, with corresponding rational parametric equations $x = p(s,t)/q(s,t)$, $y = k(s,t)/l(s,t)$, and $z = m(s,t)/n(s,t)$. Advantages of parametric equations also accrue from the fact that these forms are the adequate description for drawing curves on a plotter or graphics display screen and that the two parametric variables of surfaces also supply the coordinate grid on which embedded curves may be defined. They also prove useful in obtaining efficient algorithms for generating, with rational parametric representations, the surface boundary of configuration space obstacles arising from the collision free motion of quadric objects amongst physical obstacles, defined by patches of quadric surfaces, [3].

Both conics and conicoids always have a rational parameterization. In § 2 and § 3 of this paper we describe algorithms to obtain rational parametric equations for the conics and conicoids, given the implicit equations. Polynomial parameterizations are also obtained whenever they exist for the conics and conicoids. These parameterizations are over the field of Reals, or the field of Complex numbers when real solutions do not exist. Further in § 4 we consider the rational parameterizations over the fields of rationals, reals and complex numbers. Cubics (degree 3) plane curves and cubicoids (degree 3) surfaces do not always have a rational parameterization. However they always have a parameterization of the type which allows a single square root of rational functions. In a subsequent paper [2], we show how to obtain the rational and special parametric equations for cubics and cubicoids.

1. Introduction

General curves and surfaces can be represented by parametric or implicit equations. For various reasons related to efficient computability, (greater freedom in controlling shape and direct ease in performing transformations), the dominant means of representing curves and surfaces in geometric modeling is the parametric equation [4]. A general (degree two) conic implicit equation is given by $I(x,y) = ax^2 + by^2 + cxy + dx + ey + f = 0$, and rational parametric equations given by $x = p(t)/q(t)$ and $y = k(t)/l(t)$, where p, q, k and l are univariate polynomials. Further a general (degree two) conicoid implicit equation is given by $I(x,y,z) = ax^2 + by^2 + cz^2 + dxy + exz + fyz + gx + hy + iz + j = 0$, with corresponding rational parametric equations $x = p(s,t)/q(s,t)$, $y = k(s,t)/l(s,t)$, and $z = m(s,t)/n(s,t)$. Advantages of parametric equations also accrue from the fact that these forms are the adequate description for drawing curves on a plotter or graphics display screen and that the two parametric variables of surfaces also supply the coordinate grid on which embedded curves may be defined. They also prove useful in obtaining efficient algorithms for generating, with rational parametric representations, the surface boundary of configuration space obstacles arising from the collision free motion of quadric objects amongst physical obstacles, defined by patches of quadric surfaces, [3].

Both conics and conicoids always have a rational parameterization. In § 2 and § 3 of this paper we describe algorithms to obtain rational parametric equations for the conics and conicoids, given the implicit equations. Polynomial parameterizations are also obtained whenever they exist for the conics and conicoids. These parameterizations are over the field of Reals, or the field of Complex numbers when real solutions do not exist. Further in § 4 we consider the rational parameterizations over the fields of rationals, reals and complex numbers. Cubics (degree 3) plane curves and cubicoids (degree 3) surfaces do not always have a rational parameterization. However they always have a parameterization of the type which allows a single square root of rational functions. In a subsequent paper [2], we show how to obtain the rational and special parametric equations for cubics and cubicoids.

2. Conics

The general conic implicit equation is given by $I(x,y) = ax^2 + by^2 + cxy + dx + ey + f = 0$. The non-trivial case is when both a and b are non-zero. Otherwise one already has one variable in linear form and expressible as a rational polynomial expression of the other, and hence a rational parameterization. To obtain the rational parameterization all we need to do is to make $I(x,y)$ *non-regular* in x or y. That is, eliminate the x^2 or the y^2 term through a coordinate transformation. For then one of the variable is again in linear form and is expressible as a rational polynomial expression of the other. We choose to eliminate the y^2 term, by an appropriate coordinate transformation applied to $I(x,y)$. This is always possible and the algorithm is now described below. (The entire algorithm which also handles all trivial and degenerate cases of the conic is implemented on a VAX-780 using VAXIMA, a listing of which is provided in the appendix.)

(1) If $I(x,y)$ has a real root at infinity, a *linear transformation* of the type $x = a_1x + b_1y + c_1$ and $y = a_2x + b_2y + c_2$ will suffice. If $I(x,y)$ has no real root at infinity, we must use a *fractional linear transformation* of the type $x = (a_1x + b_1y + c_1)/(a_3x + b_3y + c_3)$ and $y = a_2x + b_2y + c_2)/(a_3x + b_3y + c_3)$. This is equivalent to a *homogeneous linear transformation* of the type $X = a_1X + b_1Y + c_1Z$, $Y = a_2X + b_2Y + c_2Z$ and $Z = a_3X + b_3Y + c_3Z$ applied to the homogeneous conic $I(X,Y,Z) = aX^2 + bY^2 + cXY + dXZ + eYZ + fZ^2 = 0$.

(2) Points at infinity for $I(x,y)$ correspond to linear factors of the *degree form* (highest degree terms) of I. For the conic this corresponds to a real root at infinity if $c^2 \geq 4ab$. For otherwise both roots at infinity are complex , (complex roots arise in conjugate pairs). Further $c^2 = 4ab$ corresponds to a polynomial parameterization for the conic, as then the degree form is a perfect square.

(3) Applying a linear transformation for $c^2 \geq 4ab$, gives rise to $I(x,y) = I(a_1x + b_1y + c_1, a_2x + b_2y + c_2)$. To eliminate the y^2 term we need to choose b_1 and b_2 such that $ab_1^2 + cb_1b_2 + bb_2^2 = 0$. Here both the values of b_1 and b_2 can always be chosen to be real.

(4) Applying a homogeneous linear transformation for $c^2 < 4ab$, gives rise to $I(X,Y,Z) = I(a_1X + b_1Y + c_1Z, a_2X + b_2Y + c_2Z, a_3X + b_3Y + c_3Z)$. To eliminate the Y^2 term we need to choose b_1, b_2 and b_3 such that $ab_1^2 + bb_2^2 + cb_1b_2 + db_1b_3 + eb_2b_3 + fb_3^2 = 0$. This is equivalent to finding a point (b_1, b_2, b_3) on the homogeneous conic. The values of b_1 and b_2 are both real if $(cd - 2ae)$ is not less than the *geometric mean* of $4af - d^2$ and $4ab - c^2$.

(5) Finally choose the remaining coefficients a_i's, c_i's, ensuring that the appropriate transformation is well defined. In the case of a linear transformation, this corresponds to ensuring that the matrix $\begin{vmatrix} a_1 & b_1 \\ a_2 & b_2 \end{vmatrix}$ is non-singular. Hence c_i's can be chosen to be 0 and $a_1 = 1$, $a_2 = 0$. In the case of a homogeneous linear transformation, one needs to ensure that the matrix $\begin{vmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \end{vmatrix}$ is non-singular. Here $a_1 = 1$, $c_2 = 1$ and the rest set to 0 suffices.

3. Conicoids

The case of the conicoid is a generalization of the method of the conic. The general conicoid implicit equation is given by $I(x,y,z) = ax^2 + by^2 + cz^2 + dxy + exz + fyz + gx + hy + iz + j = 0$. Again the main case of concern is when a, b and c are all non-zero. Otherwise one already has one variable in linear form and expressible as a rational polynomial expression of the other two, and hence a rational parameterization. To obtain the rational parameterization all we need to do again is to make $I(x,y,z)$ *non-regular* in say, y. That is, eliminate the y^2 term through a coordinate transformation. For then y is in linear form and is expressible as a rational polynomial expression of the other two. We eliminate the y^2 term by an appropriate coordinate transformation applied to $I(x,y,z)$. This is always possible and the algorithm is now described below. (The entire algorithm which also handles all trivial and degenerate cases of the conicoid is implemented on a VAX-780 using VAXIMA, a listing of which is provided in the appendix.)

(1) If $I(x,y,z)$ has a real root at infinity, a *linear transformation* of the type $x = a_1x + b_1y + c_1z + d_1$, $y = a_2x + b_2y + c_2z + d_2$ and $z = a_3x + b_3y + c_3z + d_3$ will suffice. If $I(x,y,z)$ has no real root at infinity, we must use a *fractional linear*

(4) Applying a homogeneous linear transformation for $c^2 < 4ab$, gives rise to $I(X,Y,Z) = I(a_1X + b_1Y + c_1Z, a_2X + b_2Y + c_2Z, a_3X + b_3Y + c_3Z)$. To eliminate the Y^2 term we need to choose b_1, b_2 and b_3 such that $ab_1^2 + bb_2^2 + cb_1b_2 + db_1b_3 + eb_2b_3 + fb_3^2 = 0$. This is equivalent to finding a point (b_1, b_2, b_3) on the homogeneous conic. The values of b_1 and b_2 are both real if $(cd-2ae)$ is not less than the *geometric mean* of $4af - d^2$ and $4ab - c^2$.

(5) Finally choose the remaining coefficients a_i's, c_i's, ensuring that the appropriate transformation is well defined. In the case of a linear transformation, this corresponds to ensuring that the matrix $\begin{vmatrix} a_1 & b_1 \\ a_2 & b_2 \end{vmatrix}$ is non-singular. Hence c_i's can be chosen to be 0 and $a_1 = 1$, $a_2 = 0$. In the case of a homogeneous linear transformation, one needs to ensure that the matrix $\begin{vmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \end{vmatrix}$ is non-singular. Here $a_1 = 1$, $c_2 = 1$ and the rest set to 0 suffices.

3. Conicoids

The case of the conicoid is a generalization of the method of the conic. The general conicoid implicit equation is given by $I(x,y,z) = ax^2 + by^2 + cz^2 + dxy + exz + fyz + gx + hy + iz + j = 0$. Again the main case of concern is when a, b and c are all non-zero. Otherwise one already has one variable in linear form and expressible as a rational polynomial expression of the other two, and hence a rational parameterization. To obtain the rational parameterization all we need to do again is to make $I(x,y,z)$ *non-regular* in say, y. That is, eliminate the y^2 term through a coordinate transformation. For then y is in linear form and is expressible as a rational polynomial expression of the other two. We eliminate the y^2 term by an appropriate coordinate transformation applied to $I(x,y,z)$. This is always possible and the algorithm is now described below. (The entire algorithm which also handles all trivial and degenerate cases of the conicoid is implemented on a VAX-780 using VAXIMA, a listing of which is provided in the appendix.)

(1) If $I(x,y,z)$ has a real root at infinity, a *linear transformation* of the type $x = a_1x + b_1y + c_1z + d_1$, $y = a_2x + b_2y + c_2z + d_2$ and $z = a_3x + b_3y + c_3z + d_3$ will suffice. If $I(x,y,z)$ has no real root at infinity, we must use a *fractional linear*

transformation of the type $x = (a_1x + b_1y + c_1z + d_1)/(a_4x + b_4y + c_4z + d_4)$, $y = a_2x + b_2y + c_2z + d_2)/(a_4x + b_4y + c_4z + d_4)$. and $z = (a_3x + b_3y + c_3z + d_3)/(a_4x + b_4y + c_4z + d_4)$. This is equivalent to a *homogeneous linear transformation* of the type $X = a_1X + b_1Y + c_1Z + d_1W$, $Y = a_2X + b_2Y + c_2Z + d_2W$, $Z = a_3X + b_3Y + c_3Z + d_3W$ and $W = a_4X + b_4Y + c_4Z + d_4W$ applied to the homogeneous conicoid $I(X,Y,Z,W) = aX^2 + bY^2 + cZ^2 + dXY + eXZ + fYZ + gXW + hYW + iZW + jW^2 = 0$,

(2) Points at infinity for $I(x,y)$ correspond to linear factors of the *degree form* (highest degree terms) of I. For the conicoid this corresponds to the roots of the homogeneous conic equation $C(x,y,z) = ax^2 + by^2 + dxy + exz + fyz + cz^2 = 0$. Also, here the simultaneous truth of $d^2 = 4ab$, $e^2 = 4ac$ and $f^2 = 4bc$ corresponds to the existence of a polynomial parameterization for the conicoid, as then the degree form is a perfect square.

(3) Apply a linear transformation if a real root (r_x, r_y, r_z) exists for the homogeneous conic $C(x,y,z)$ of (2). This gives rise to $I(x,y,z) = I(\ a_1x + b_1y + c_1z + d_1$, $a_2x + b_2y + c_2z + d_2$, $a_3x + b_3y + c_3z + d_3)$. To eliminate the y^2 term we can take $(b_1, b_2, b_3) = (r_x, r_y, r_z)$, the real point on $C(x,y,z)$.

(4) Apply a homogeneous linear transformation if only complex roots exist for the homogeneous conic $C(x,y,z)$ of (2). This gives rise to $I(X,Y,Z,W) = I(\ a_1X + b_1Y + c_1Z + d_1W$, $a_2X + b_2Y + c_2Z + d_2W$, $a_3X + b_3Y + c_3Z + d_3W$, $a_4X + b_4Y + c_4Z + d_4W)$. To eliminate the Y^2 term we choose $b_4 = 1$ and (b_1, b_2) to be a point on either the conic $ax^2 + by^2 + dxy + gxz + hyz + jz^2 = 0$ with $b_3 = 0$ or a point on the conic $ax^2 + by^2 + dxy + (e+g)xz + (f+h)yz + (c+i+j)z^2 = 0$ with $b_3 = 1$. Real values exist for b_1 and b_2 if there exists a real point on either of the above conics.

(5) Finally choose the remaining coefficients a_i's, c_i's, and d_i's, ensuring that the appropriate transformation is well defined. In the case of a linear transformation, this corresponds to ensuring that the matrix $\begin{vmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \end{vmatrix}$ is non-singular. Here the d_i's can be chosen to be 0.

Further $a_2 = 1$, $c_3 = 1$ if b_1 is non-zero or else $a_1 = 1$, $c_3 = 1$ if b_2 is non-zero or else $a_1 = 1$, $c_2 = 1$, with the rest set to 0. In the case of a homogeneous linear transformation one needs to ensure that the matrix $\begin{vmatrix} a_1 & b_1 & c_1 & d_1 \\ a_2 & b_2 & c_2 & d_2 \\ a_3 & b_3 & c_3 & d_3 \\ a_4 & b_4 & c_4 & d_4 \end{vmatrix}$ is non-singular. Here $a_1 = 1$, $c_3 = 1$, $d_2 = 1$ with the rest set to 0 suffices.

4. Rational Fields

As seen from the previous sections one obtains parameterizations over the reals or the complex numbers if the corresponding coefficients of the appropriate transformations are over the fields of reals or complex numbers respectively. The coefficients themselves correspond to finding real or complex points on various conic equations. Thus essentially the question of whether the parameterization for conics and conicoids is possible over the field of rationals, reduces to the question of whether there exists a rational root of a certain conic equation with integral coefficients, (or an integral root of the homogenized conic equation). The answer to the latter question is given by an existence criterion in [5]. When such an integral root exists one can obtain it by solving an appropriate diophantine equation of the type $x^2 - D{*}y^2 = N$, for integer D and N, (a diophantine equation which has come to be known as Pell's equation, though could also be called Bhaskara's equation [1]).

To compute an integral point on a homogeneous conic, $C(X,Y,Z) = aX^2 + bY^2 + cXY + dXZ + eYZ + fZ^2 = 0$, one could find the point at infinity ($Z = 0$), or at finite distances ($Z = 1$). Such a point exists at infinity if $c^2 - 4ab$ is a perfect square. Finding integral points at finite distances is equivalent to finding rational points of the dehomogeneized conic $C(X,Y,1)$. This corresponds to finding a rational solution (b_1,b_2) of the equation, $ab_1^2 + (cb_2+d)b_1 + (bb_2^2+eb_2+f) = 0$. Such a solution exists when the discriminant of the equation is a perfect square, (equal to y^2 for integer y). This reduces to finding a rational solution of the equation, $(c^2-4ab)b_2^2 + 2(cd-2ae)b_2 + (d^2-4af-y^2) = 0$, where such a solution again exists when its discriminant is a perfect square, (equal to x^2 for integer x). Hence we need to solve the equation $x^2 - D{*}y^2 = N$, for diophantine solutions x and y, with $D = c^2 - 4ab$ and $N = (cd - 2ae)^2 - (c^2 - 4ab)(d^2 - 4af)$. If D is negative or a perfect square there are only a finite number of solutions to this equation. If D is positive, solutions can be obtained by simple continued fractions.

To compute an integral point on a homogeneous conicoid, $I(X,Y,Z,W) = aX^2 + bY^2 + cZ^2 + dXY + eXZ + fYZ + gXW + hYW + iZW + jW^2 = 0$, one could again find the point at infinity $(Z = 0)$, or at finite distances $(Z = 1)$. Finding an integral point at infinity reduces to the above case of finding an integral root of a homogeneous conic $aX^2 + bY^2 + dXY + eXZ + fYZ + cZ^2 = 0$. Finding integral points at finite distances also reduces to the earlier case of solving for a rational point of a conic $ax^2 + by^2 + dxy + gx + hy + j^2 = 0$, or a rational point of the conic $ax^2 + by^2 + dxy + (e+g)x + (f+h)y + (c+i+j)^2 = 0$.

5. References

[1] Abhyankar, S. S., *Historical Ramblings in Algebraic Geometry and Related Algebra*, American Mathematical Monthly, vol 83, no 6, (June 1976), p409-448

[2] Abhyankar, S. S., and Bajaj, C., *Automatic Rational Parameterization of Curves and Surfaces II: Cubics and Cubicoids*, (in preparation).

[3] Bajaj, C., and Kim, M., *Generation of Configuration Spaces II: The Case of Moving Quadrics*, Computer Science Technical Report, CSD-TR-586, Purdue University, (March 1986)

[4] Mortenson, M., *Geometric Modeling*, John Wiley & Sons, Inc., 1985.

[5] Smith, H.J., *On the Criterion of Resolubility in Integral Numbers of the Indeterminate Equation*, Proc. of the Royal Society, vol. xiii, p110 -111, 1864.

Appendix: A Listing of Vaxima Code

6. Curve Real Complex Point (CRCPNT)

```
        crcpnt(a,b,c,d,e,f) :=
  block([b1,b2,b3],
      if c^2 >= 4 * a * b then /* Real point at infinity */
      block(
            b3 : 0,
            ds  : sqrt(c^2 - 4 * a * b),
            s1 : (-c + ds) / (2*a),
            s2 : (-c - ds) / (2*a),
            if c < 0 then b1 : s1
            else b1 : s2,
            b2 : 1
         )
      else /* point at finite distance */
        block(
            b3 : 1,
            if (c*d -2*a*e)^2 < (4*a*f-d^2)(4*a*b-c^2)
            then print("sorry complex"),
            b1 : -(c*b2 +d)/2*a
            b2 : ((c*d-2*a*e) + sqrt((c*d-2*a*e)^2 -(4*a*b-c^2)(4*a*f-d^2)))/(2*a),
         ),
      [b1,b2,b3]
     )$
```

7. Curve Implicit To Parametric (CITOP)

```
      citop (poly, xp, yp, flag) :=
block ([a, b, c, d, e, f, x, y, b1, b2, b3],

/*
** Calculate and print coefficients of poly.
*/
       a : ratcoef(poly, x^2, 1),
       b : ratcoef(poly, y^2, 1),
      c : ratcoef(poly, x*y, 1),
       d : ratcoef(ratcoef(poly, x, 1), y, 0),
       e : ratcoef(ratcoef(poly, y, 1), x, 0),
       f : ratcoef(ratcoef(poly, y, 0), x, 0),
       print("a =", a), print("b =", b), print("c =", c),
       print("d =", d), print("e =", e), print("f =", f),

/*
** Determine which case we need to handle
*/
       if a < 0 then block(a:-a,b:-b,c:-c,d:-d,e:-e,f:-f),

       if a # 0 and b # 0 then /* make non-regular in x or y */
              if (c^2 - 4*a*b) = 0 and d = 0 and e = 0 then
                block(
                       if f > 0 then print("sorry  complex"),
                       if c < 0 then sgn : 1 else sgn : -1,
                       x : sgn*sqrt(b/a)*t + sqrt(-f/a),
                       y : t
```

```
          )
        else
         block(
          x  : crcpnt(a,b,c,d,e,f),
          b1 : part(x,1), b2 : part(x,2), b3 : part(x,3),
          if b3 = 0 then /* linear transformation */
           block(
                  y : - (a*t^2 + d*t + f)/
                       ((2*a*b1 + c*b2)*t + (d*b1 +e*b2)),
                  x :  ((a*b1 +c*b2)*t^2 +e*b2*t -f*b1)/
                       ((2*a*b1 + c*b2)*t + (d*b1 +e*b2))
                 )
          else /* homogeneous linear transformation */
           block(
              if flag # 0 then block( b1 : xp, b2 : yp ),
                x : -((a*b1 + c*b2 + d*b3)*t^2  + (2*b*b2 + e*b3)*t -b*b1)/
                   (b3*(a*t^2+c*t+b)),

                y : (a*b2*t^2 - (2*a*b1 + d*b3)*t - (b*b2 + c*b1 + e*b3))/
                    (b3*(a*t^2+c*t+b))
            )
          )
        else /* already non-regular in y */
         if a # 0 then
                  if c = 0 and e = 0 then
                    block(
                         if d^2 - 4*a*f < 0 then print("sorry complex"),
                         x: (-d-sqrt(d^2-4*a*f))/(2*a),
                         y: t
                         )
                  else
                  block(
                         y : - (a*t^2 + d*t + f) / (c*t + e),
                         x : t
                  )
         else /* already non-regular in x */
              if b # 0 then
                  if c = 0 and d = 0 then
                         block(
                                if e^2 - 4*b*f < 0 then print("sorry complex"),
                                y: (-e-sqrt(e^2-4*b*f))/(2*b),
                                x: t
                              )
                  else
                         block(
                                x : - (b*t^2 + e*t + f) / (c*t + d),
                                y : t
                              )
              else
                  if c # 0 or d # 0 then
                         block(
                                x: -(e*t + f)/(c*t + d),
                                y: t
                              )
                  else
                    if e # 0 then block( y: -f/e , x: t)
                    else  block( print("constant value"), x:t, y:t),
print("eval=", ratsimp(a*x^2 +b*y^2 +c*x*y +d*x +e*y +f)),
 [x,y]
```

```
)$
```

8. Surface Real Complex Point (SRCPNT)

```
        srcpnt(a,b,c,d,e,f,g,h,i,j) :=
  block([b1,b2,b3,b4,x],

      if (d^2 >= 4 * a * b) or ((d*e -2*a*f)^2 >= (4*a*c - e^2)(4*a*b - d^2)) then
      block( /* Real point at infinity */
            b4 : 0,
            x  : crcpnt(a,b,d,e,f,c),
            b1 : part(x,1), b2 : part(x,2), b3 : part(x,3)
         )
      else
         block( /* point at finite distance */
                  b4 : 1,
                  if ((d*g -2*a*h)^2 >= (4*a*j-g^2)(4*a*b -d^2)) then
                  block(
                        x  : crcpnt(a,b,d,g,h,j),
                        b1 : part(x,1), b2 : part(x,2), b3 : 0
                     )
                  else
                  block(
                        x  : crcpnt(a,b,d,(e+g),(f+h),(c+i+j)),
                        b1 : part(x,1),
                        b2 : part(x,2),
                        b3 : 1
                     )
            ),
     [b1,b2,b3,b4]
    )$
```

9. Surface Implicit To Parametric (SITOP)

```
      sitop (poly, xp, yp, zp, flag) :=
block ([a, b, c, d, e, f, g, h, i, j, x, y, z, b1, b2, b3, b4],

/*
** Calculate and print coefficients of poly.
*/
      a : ratcoef(poly, x^2, 1),
      b : ratcoef(poly, y^2, 1),
      c : ratcoef(poly, z^2, 1),
      d : ratcoef(poly, x*y, 1),
      e : ratcoef(poly, x*z, 1),
      f : ratcoef(poly, y*z, 1),
      g : ratcoef(poly, x, 1) - d*y - e*z,
      h : ratcoef(poly, y, 1) - d*x - f*z,
      i : ratcoef(poly, z, 1) - e*x - f*y,
      j : ratcoef(ratcoef(ratcoef(poly, x, 0), y, 0), z , 0),
      print("a =", a), print("b =", b), print("c =", c), print("d =", d), print("e =", e),
      print("f =", f), print("g =", g), print("h =", h), print("i =", i), print("j =", j),

/*
** Determine which case we need to handle
*/
      a1 : 0, a2 : 0, a3 : 0,
      c1 : 0, c2 : 0, c3 : 0,
      d1 : 0, d2 : 0, d3 : 0,
```

```
if a < 0 then block( a:-a,b:-b,c:-c,d:-d,e:-e,f:-f,g:-g,h:-h,i:-i,j:-j),

if a # 0 and b # 0 and c # 0 then /* make non-regular in x or y or z */
 block(
     cnd1 : 0,
     if d > 0 then if e > 0 and f > 0 then block( sgn1:-1, sgn2:-1)
              else if e < 0 and f < 0
                   then block( sgn1:-1, sgn2:1) else cnd1:1
     else if e > 0 and f < 0 then block( sgn1:1, sgn2:-1)
              else if e < 0 and f > 0
                   then block( sgn1:1, sgn2:1) else cnd1:1,
     cnd2 : (d^2 = 4*a*b) and (e^2 = 4*a*c) and (f^2 = 4*b*c),

     if cnd1 = 0 and cnd2 and g = 0 and h = 0 and i = 0 then
      block(
           if j > 0 then print("sorry  complex"),
           x : sgn1*sqrt(b/a)*t + sgn2*sqrt(c/a)*s + sqrt(-j/a),
           y : t,
           z : s
        )
     else
      block(
        x : srcpnt(a,b,c,d,e,f,g,h,i,j),
        b1 : part(x,1), b2 : part(x,2), b3 : part(x,3), b4 : part(x,4),
        if b4 = 0 then /* linear transformation */
          block(
                   if b1 # 0 then block( a2 : 1, c3 : 1)
                   else if b2 # 0 then block( a1 : 1, c3 : 1)
                      else block( a1 : 1, c2 : 1),

                   yy : -((a*a1^2 +b*a2^2 +c*a3^2 +d*a1*a2
                           +e*a1*a3 +f*a2*a3)*t^2
                        +(a*c1^2 +b*c2^2 +c*c3^2 +d*c1*c2
                           +e*c1*c3 +f*c2*c3)*s^2
                        +(d*a1*c2 +d*c1*a2 +e*a1*c3 +e*c1*a3
                          +f*a2*c3 +f*c2*a3)*s*t
                        +(g*a1 +h*a2 +i*a3)*t +(g*c1 +h*c2 +i*c3)*s
                        +j) /
                        ((2*a*a1*b1 +2*b*a2*b2 +2*c*a3*b3 +d*a1*b2
                           +d*b1*a2 +e*a1*b3 +e*b1*a3 +f*a2*b3 +f*b2*a3)*t
                        +(2*a*b1*c1 +2*b*b2*c2 +2*c*b3*c3 +d*c1*b2
                           +d*b1*c2 +e*c1*b3 +e*b1*c3 +f*c2*b3 +f*b2*c3)*s
                        + g*b1 +h*b2 +i*b3),

                   x : ratsimp(a1*t + b1*yy + c1*s),
                   y : ratsimp(a2*t + b2*yy + c2*s),
                   z : ratsimp(a3*t + b3*yy + c3*s)
                )
        else /* homogeneous linear transformation */
         block(a1 : 1, c3 : 1, d2 : 1,
             if flag # 0 then block( b1 : xp, b2 : yp, b3 : zp),

                   yy : -(a*(a1*t +c1*s +d1)^2
                        +b*(a2*t +c2*s +d2)^2
                        +c*(a3*t +c3*s +d3)^2
                        +d*(a1*t +c1*s +d1)*(a2*t +c2*s +d2)
                        +e*(a1*t +c1*s +d1)*(a3*t +c3*s +d3)
                        +f*(a2*t +c2*s +d2)*(a3*t +c3*s +d3)
```

```
                        ) /
                        ((2*a*b1 +d*b2 +e*b3)*(a1*t +c1*s +d1)
                        +(2*b*b2 +d*b1 +f*b3)*(a2*t +c2*s +d2)
                        +(2*c*b3 +e*b1 +f*b2)*(a3*t +c3*s +d3)),

                      x : ratsimp((t + b1*yy)/yy),
                      y : ratsimp((1 + b2*yy)/yy),
                      z : ratsimp((s + b3*yy)/yy)
               )
             )
         )
else
 if a = 0 then /* already non-regular in x */
      if d = 0 and e = 0 and g = 0 then
        block(
             print("conic"),
             xx : citop(b*x^2 + c*y^2 + f*x*y + h*x + i*y +j,0,0,0),
              x : s,
              y : part(xx,1),
              z : part(xx,2)
             )
      else
        block(
             x : - (b*t^2 +c*s^2 +f*s*t +h*t + i*s + j) / (d*t +e*s + g),
             y : t,
             z : s
             )
else
   if b = 0 then /* already regular in y */
     if d = 0 and f = 0 and h = 0 then
        block(
             print("conic"),
             xx : citop(a*x^2 + c*y^2 +e*x*y +g*x +i*y +j,0,0,0),
              y : s,
              x : part(xx,1),
              z : part(xx,2)
             )
      else
        block(
             y : - (a*t^2 +c*s^2 +e*t*s +g*t + i*s + j) / (d*t +f*s + h),
             x : t,
             z : s
             )
  else /* already regular in z */
      if e = 0 and f = 0 and i = 0 then
        block(
             print("conic"),
             xx : citop(a*x^2 + b*y^2 +d*x*y +g*x +h*y +j,0,0,0),
              z : s,
              x : part(xx,1),
              y : part(xx,2)
             )
      else
        block(
             z : - (a*t^2 +b*s^2 +d*s*t +g*t + h*s + j) / (e*t +f*s + i),
             x : t,
             y : s
             ),
```

```
    print("eval=", ratsimp(a*x^2 +b*y^2 +c*z^2 +d*x*y +e*x*z +f*y*z +g*x +h*y +i*z +j)),
     [x,y,z]
)$
```